Release J: rApp Manager

- Introduction
- Architecture
- rApp Data Model
- Integrations
    - ACM
    - DME
    - SME
- State Transitions
    - rApp States
    - rApp Instance States
- Flows
    - rApp flow
        - Create rApp
        - Delete rApp
    - rApp Instance flow
        - Create rApp Instance
        - Deploy rApp Instance
        - Undeploy rApp Instance
        - Delete rApp Instance
- Sample rApp package structure
    - ACM (Files/Acm)
    - DME (Files/Dme)
    - SME (Files/Sme)
- CSAR File Generation
- Deployment Instructions
    - Pre-requisites
    - Environment setup
    - Installation
        - Installed Components
            - Istio components
            - Cert Manager components
            - Kserve components
            - ACM Components
            - NONRTRIC Components
        - Troubleshooting
    - Uninstallation

Introduction

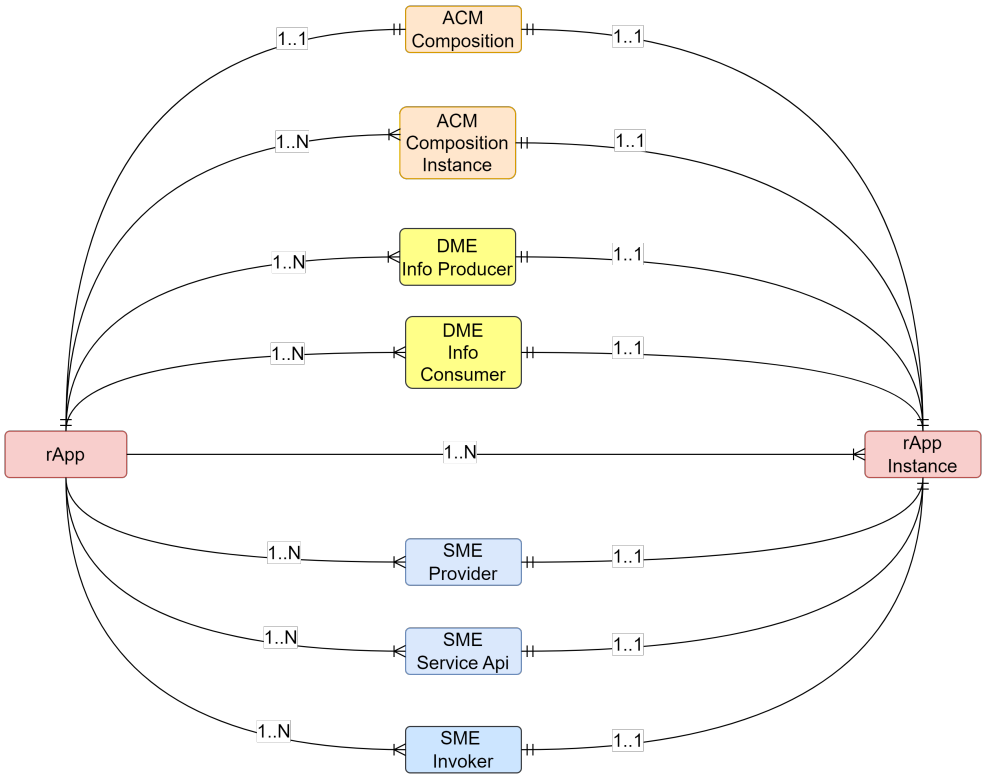
The rApp Manager is a lifecycle management service for rApps. It gets the rApp as an ASD formatted package and lifecycle manages it based on it instance configuration. It uses ONAP ACM for lifecycle management operations and it integrates with other components for managing the rApp.

The ASD package contains the details required to create and integrate the required services/components. Each ASD package contains only one rApp and one rApp can have any number of rApp instances.

Source code repository : https://gerrit.o-ran-sc.org/r/gitweb?p=nonrtric%2Fplt%2Frappmanager.git;a=summary

Architecture

rApp Data Model



Integrations

The rApp Manager is integrated with the following components to support lifecycle managing the rApp.

ACM

Automation Composition Management (ACM) is a framework that supports Life Cycle Management of Automation Compositions. It supports deployment, monitoring, update and removal of Automation Compositions en-bloc, allowing users to manage their features, services, and capabilities as single logical units. More details about ACM can be found here.

ACM-R has the ability to support an unlimited number of participants and all the participants can be configured through the configuration in the rApp package.

List of participants used by rApp manager sample rApp.

- A1PMS Participant - It interacts with A1PMS of NONRTRIC. It is capable of lifecycle managing A1PMS service.
- Kserve Participant - It interacts with Kserve. It is capable of lifecycle managing Kserve inference service.
- Kubernetes Participant - It interacts with Helm/Kubernetes. It is capable of lifecycle managing Helm charts. It expects the helm charts to be available in the mentioned repository as it doesn't handle the helm chart creation in the chart repository.
- DME Participant - It interacts with DME(ICS) of NONRTRIC. It is capable of lifecycle managing DME entities.

ACM composition and instance details can be provided as part of the rApp package and the package structure can be found here.

DME

The DME(Information Coordination Service (ICS)) is a generic service that maintains data subscriptions. Its main purpose is to decouple data consumers and data producers in a multi vendor environment. A data consumer does not need to know anything about the producers of the data. More details about DME can be found here.

It is integrated with rApp manager to enable the rApp to produce/consume specific type of data(Information Type in DME terms).

Information type, and Data producer/consumer information can be provided as part of rApp package and the package structure can be found here.

SME

The CAPIF stands for Common API framework and it was developed by 3GPP to enable a unified Northbound API framework across 3GPP network functions, and to ensure that there is a single and harmonized approach for API development. More details about SME can be found here.

It is integrated with rApp manager to enable the rApp to expose/access/discover endpoints.

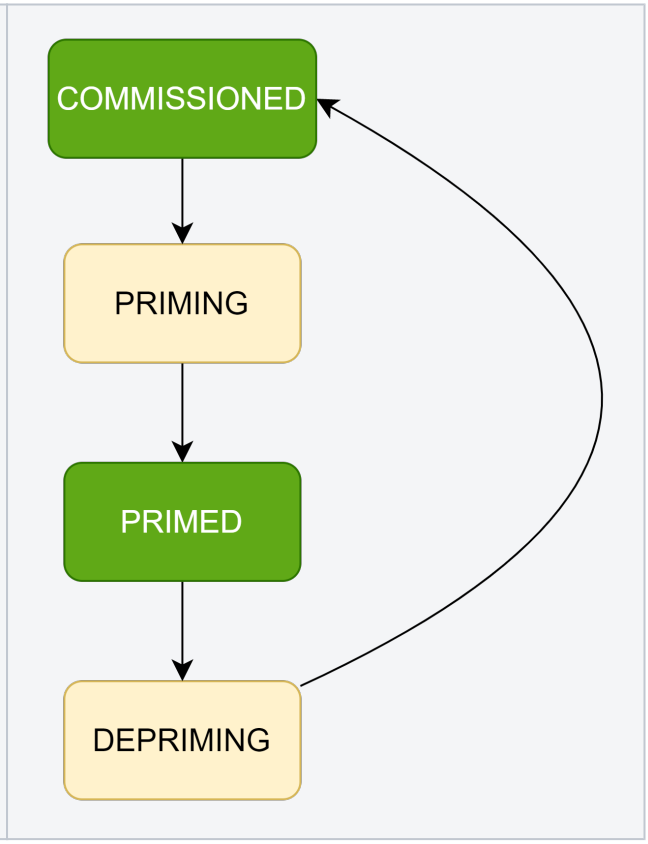
Service exposure/access related configurations can be provided as part of rApp package and the package structure can be found here.

State Transitions

rApp States

The rApp lifecycle contains 4 states. The state and transitions are as follows,

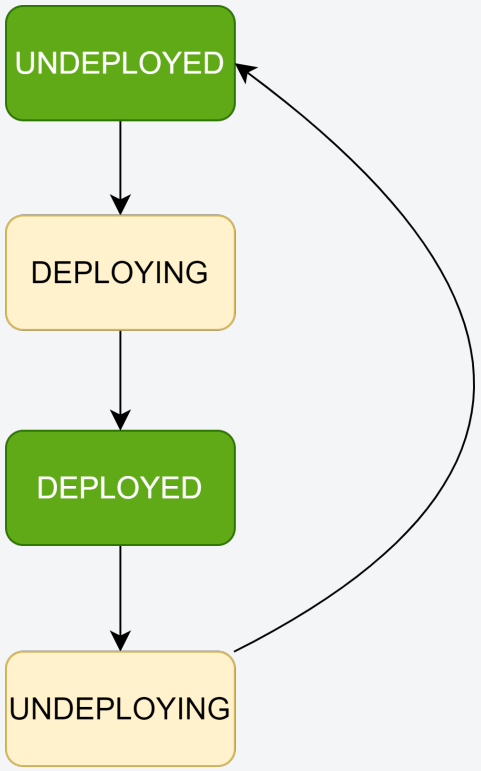
1. COMMISSIONED
   rApp get created in this state and once the DEPRIMING is completed
2. PRIMING
   This is a transition state. rApp will be in this state once the PRIMING requested for rApp
3. PRIMED
   rApp will be in this state once the PRIMING is completed. In this state rApp instances can be created
4. DEPRIMING
   This is a transition state. rApp will be in this state once the DEPRIMING requested for rApp



rApp Instance States

The rApp Instance lifecycle contains 4 states. The state and transitions are as follows,

1. UNDEPLOYED
   rApp instance gets created in this state and once the rApp Instance undeploy is completed
2. DEPLOYING
   This is the transition state. rApp instance will be in this state once DEPLOY is requested
3. DEPLOYED
   rApp instance will be in this state once the rapp instance deployment is completed.
4. UNDEPLOYING
   This is a transition state. rApp instance will be in this state once UNDEPLOY requested for rApp instance



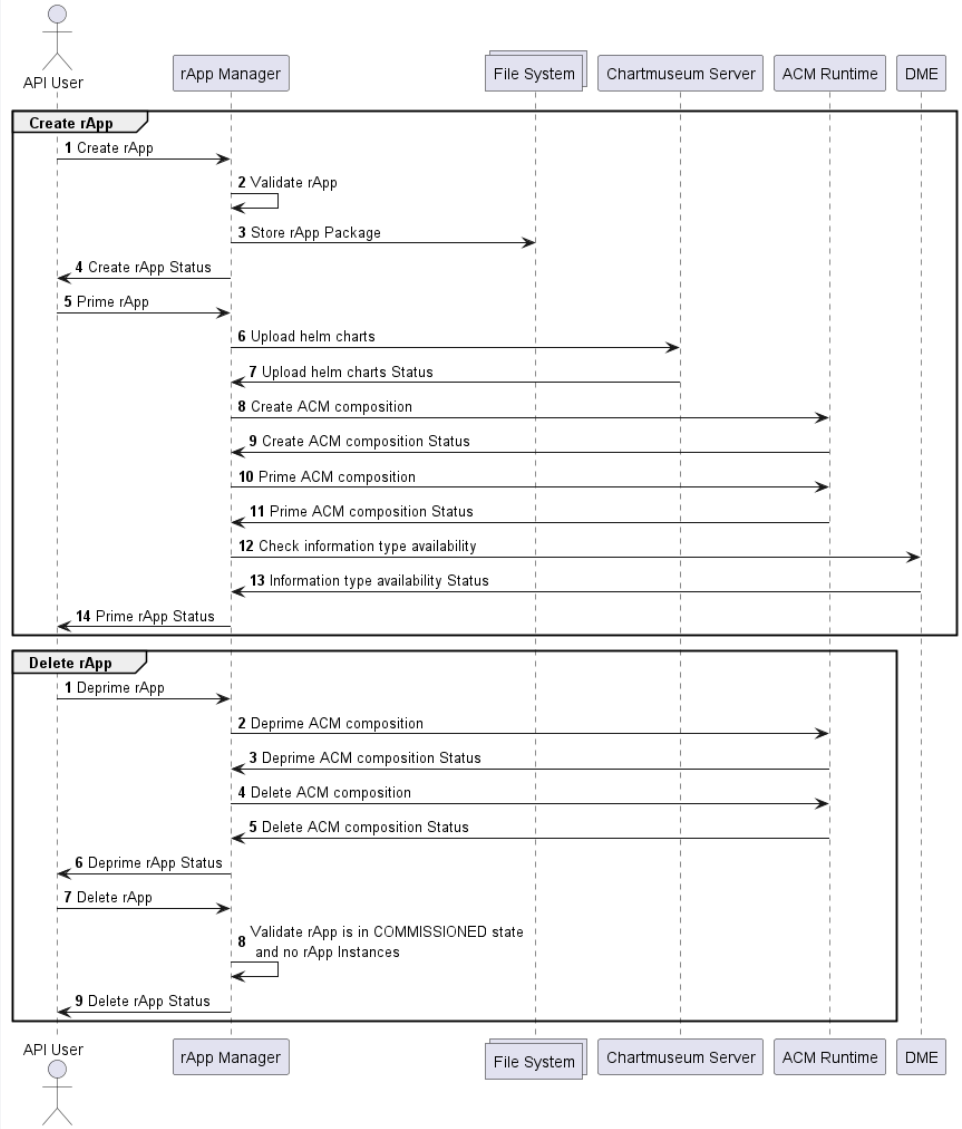
Flows

rApp flow

Create rApp	

1. API user creates rApp by sending rApp package
2. rApp Manager validates the rApp
3. rApp Manager stores the rApp in the file system if the rApp is valid
4. API user provided with the status of rApp creation.
5. API user request to Prime the rApp
6. rApp Manager uploads the helm artifacts to chart museum server
7. rApp Manager get helm artifacts upload status
8. rApp Manager fetches the ACM composition from rApp package and creates the ACM composition in ACM-R
9. rApp Manager gets the ACM composition creation status from ACM-R
10. rApp Manager request ACM-R to prime the ACM composition
11. rApp Manager gets the ACM composition priming status from ACM-R
12. rApp Manager checks with DME for the unknown information type from rApp package
13. rApp Manager get the information type availability from DME
14. API user provided with the status of rApp priming

Delete rApp



1. API user request to Deprime rApp 2. rApp Manager request ACM-R to deprime the ACM composition 3. rApp Manager get the status of ACM composition depriming. 4. rApp Manager requests ACM-R to delete the ACM composition 5. rApp Manager gets the status of ACM composition deletion 6. API user provided with the deprime rApp status. 7. API user request to delete the rApp 8. rApp Manager validates that the rApp is in COMMISSIONED state and there is no rApp Instances are available. 9. API User provided with delete rApp status	

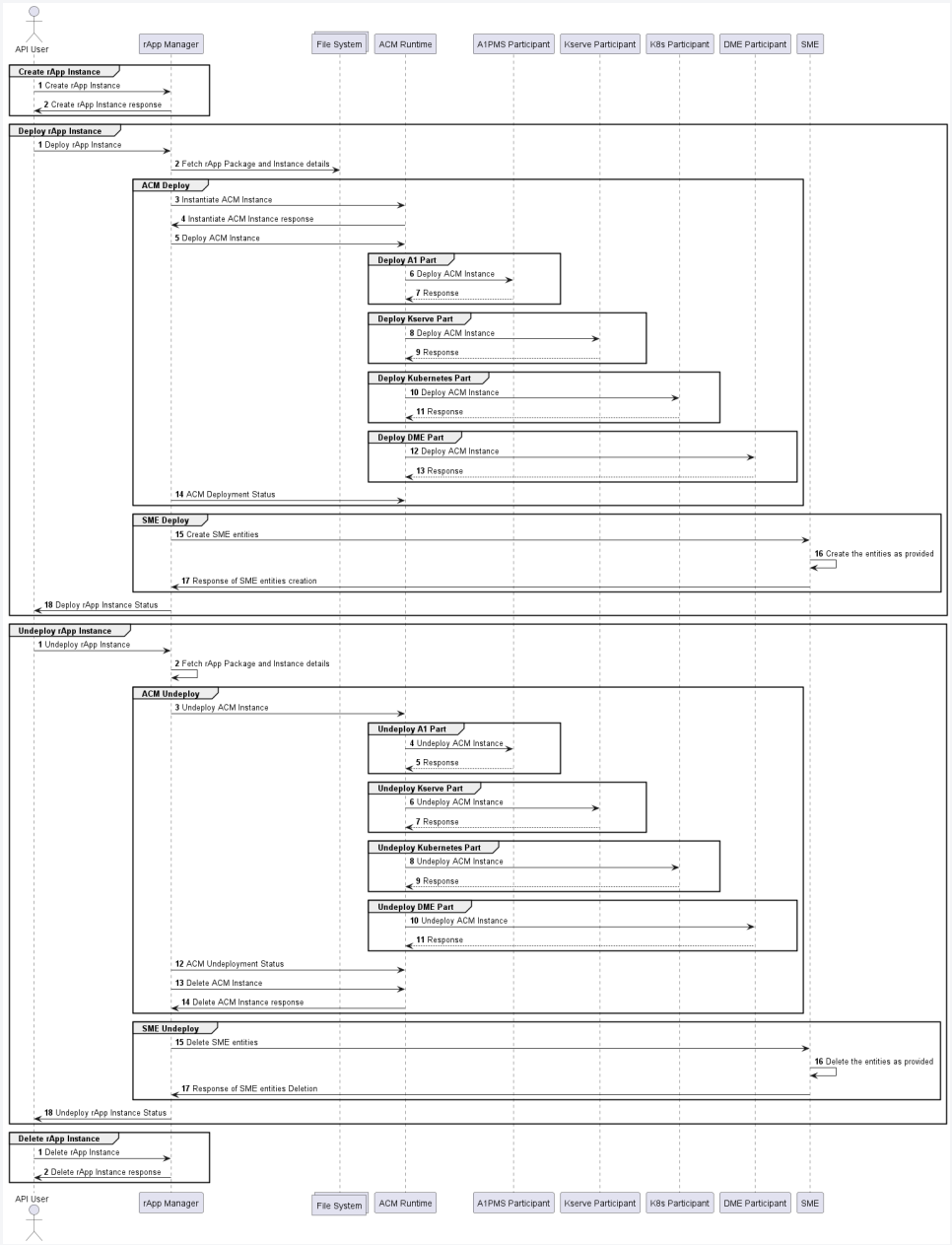
rApp Instance flow

Create rApp Instance	

1. API user request to create rApp Instance with the resource details from rApp package
2. API user get the rApp instance creation response

Deploy rApp Instance

1. API user request to deploy rApp instance
2. rApp Manager fetches the rApp package from file system and fetches the resources mentioned in the rApp instance.
3. rApp Manager request ACM-R with the necessary details from rApp package to Instantiate ACM instance



4. rApp Manager gets instantiate ACM instance response
5. rApp Manager request ACM-R to deploy ACM instance
6. ACM-R deploys the A1PMS instance if it is configured in the ACM instance
7. ACM-R gets the A1PMS deployment status
8. ACM-R deploys the Kserve instance if it is configured in the ACM instance
9. ACM-R gets the Kserve deployment status
10. ACM-R deploys the Kubernetes instance if it is configured in the ACM instance
11. ACM-R gets the Kubernetes deployment status

12. ACM-R deploys the DME instance if it is configured in the ACM instance
13. ACM-R gets the DME deployment status
14. rApp Manager gets ACM deployment status
15. rApp Manager request SME to create the entities in rApp instance
16. SME creates the entities provided by rApp Manager
17. rApp manager get the response of SME entities creation

18. API user provided with the status of rApp instance deployment. rApp Manager combines the status of ACM deployment and SME deployment to provide the rApp instance status.

Undeploy rApp Instance

1. API user request to undeploy rApp instance
2. rApp Manager fetches the rApp and rApp instance details
3. rApp Manager request ACM-R to undeploy the ACM instance
4. ACM-R undeploy the A1PMS instance if it is already deployed

5. ACM-R gets the A1PMS undeploy status
6. ACM-R undeploy the Kserve instance if it is already deployed
7. ACM-R gets the Kserve undeploy status
8. ACM-R undeploy the Kubernetes instance if it is already deployed
9. ACM-R gets the Kubernetes undeploy status
10. ACM-R undeploy the DME instance if it is already deployed
11. ACM-R gets the DME undeploy status
12. rApp Manager get ACM undeploy status
13. rApp Manager request ACM-R to delete the ACM instance

14. rApp Manager get response of ACM instance deletion
15. rApp Manager request SME to delete SME entities
16. SME deletes the entities created as part of rApp
17. rApp Manager get the response of SME entities deletion
18. API user get the status of undeploy rApp instance

Delete rApp Instance

1. API user request rApp Manager to delete rApp Instance
2. API user gets delete rApp instance response

Sample rApp package structure

This packaging structure is a prototype and subject to change

The sample rApp package structure shown below and the location of relevant files for each integration is provided.

ACM (Files /Acm)

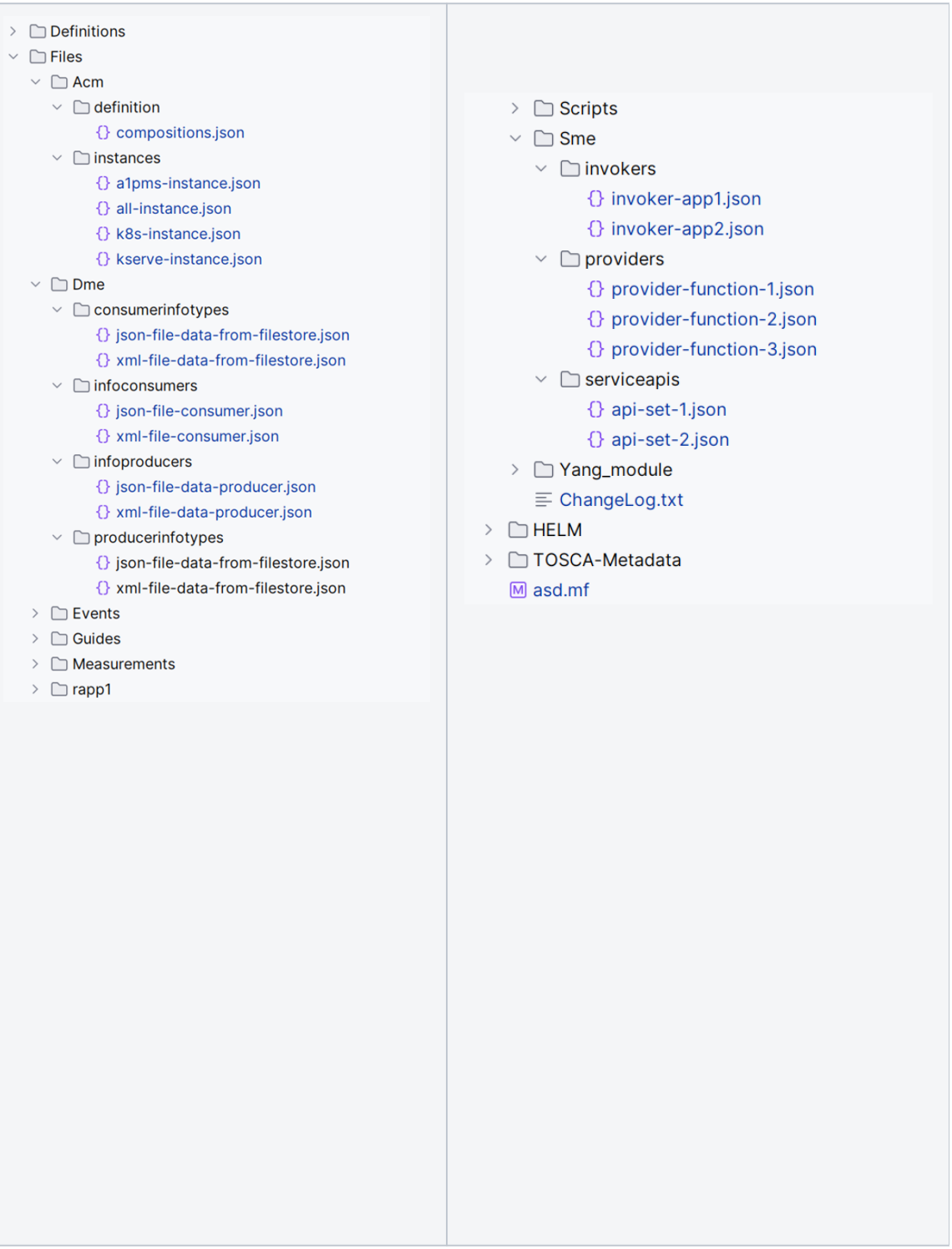
- Definition - Files/Acm /Definition (Only one file)
- Instances - Files/Acm /instances

DME (Files /Dme)

- Consumer Information Types - Files/Dme /consumerinfotypes
- Producer information Types - Files /Dme /producerinfotypes
- Information Consumers - Files/Dme /infoconsumers
- Information Producers - Files/Dme /infoproducers

SME (Files /Sme)

- Providers Functions - Files/Sme /providers
- Service Api - Files /Sme /serviceapis
- Invokers - Files/Sme /invokers



CSAR File Generation

CSAR file generator is available in the rAppmanager repository here (master).

"rappmanager/sample-rapp-generator" folder contains sample rApp packages.

The contents of the rApp (Eg. rapp-all, rapp-hello-world...) directory can be modified as required and the package can be generated as shown below,

Linux: Generate rApp package

```
> ./generate.sh <FOLDER_NAME>
```

Windows: Generate rApp package

```
> ./generate.bat <FOLDER_NAME>
```

This will generate a package named " <FOLDER_NAME>.*csar*". It can be renamed as required.

This generated package can be used with rApp Manager to create rApp.

Deployment Instructions

The scripts for the deployments of rApp Manager and its dependent components are available here (i-release).

These scripts are specifically designed for a fresh environment. Some tweaks may be required to run these in an environment where there are some existing installations

Pre-requisites

- Kubernetes Cluster (V1.24.6)
- GIT

Environment setup

The installation scripts shown here installs all the required components. It installs chart museum server where the installation script is running and it will get whitelisted in ACM. All sample rApps uses similar addresses for referring the charts in asd.yaml and Kubernetes instance configuration.

Separate chart museum can be used here and It should be whitelisted in ACM manually and the server IP/FQDN should be used in the rApp package configuration such as asd.yaml and Kubernetes instance configuration (**Chart museum server should be reachable from rApp manager and Kubernetes-Participant**).

ACM components should be configured with couple of other components for the participants to work.

In case some of the installation is already setup or not set by the installation scripts, the below environment variables can be used to set the configurations ACM through installation scripts.

Ignore the below variables if the entire environment is being setup by the following installation scripts

Variable Name	Description	Default Value
A1PMS_HOST	**Address of the A1PMS.** **It will be accessed from A1PMS participant.**	**http://policymanagementservice.nonrtric:9080**

CHART_REPO_GET_URI	**URI to get the charts.** **It will be used by Kubernetes participant and sample rApp generator.**	**http://IP_ADDRESS:8879/charts** **IP_ADDRESS: IP of the host in which the installation scripts are running.**
CHART_REPO_POST_URI	**URI to upload the charts.** **It will be used by sample rApp generator.**	**http://IP_ADDRESS:8879/charts/api/charts** **IP_ADDRESS: IP of the host in which the installation scripts are running.**

Installation

All components can be installed using the command below,

```
>./install-all.sh
```

Individual components can be installed using the commands below,

To install the tools required for other installer scripts.

```
>./install-base.sh
```

To install the ACM, and it's related components.

```
>./install-acm.sh
```

To install the Kserve, and it's related components.

```
>./install-kserve.sh
```

To installs the NONRTRIC components.

```
>./install-nonrtric.sh
```

Dev mode of installation can be done by providing an argument "**dev**" to the installation scripts above. **Eg.,** install-all.sh dev

Dev mode installation uses snapshot images for rApp manager and DME participant.

Installed Components

The below components should be up and running for the rApp Manager integrations to work properly.

Istio components

```
root@est-selfservice-booking194-k8s-master-0:/home/infra/rappmanager/scripts/install# kubectl get all -n istio-system
NAME                          READY   STATUS    RESTARTS   AGE
pod/istiod-74657777df-69lpc   1/1     Running   0          33m

NAME             TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)                                 AGE
service/istiod   ClusterIP   10.233.37.0   <none>        15010/TCP,15012/TCP,443/TCP,15014/TCP   33m

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/istiod   1/1     1            1           33m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/istiod-74657777df   1         1         1       33m

NAME                                                  REFERENCE           TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/istiod   Deployment/istiod   <unknown>/80%   1         5         1          33m
```

Cert Manager components

```
root@est-selfservice-booking194-k8s-master-0:/home/infra/rappmanager/scripts/install# kubectl get all -n cert-manager
NAME                                           READY   STATUS    RESTARTS   AGE
pod/cert-manager-7c869867bf-xjb7v              1/1     Running   0          31m
pod/cert-manager-cainjector-54c9d9b775-ndrhw   1/1     Running   0          31m
pod/cert-manager-webhook-7f7469bdb7-2rwtk      1/1     Running   0          31m

NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/cert-manager           ClusterIP   10.233.18.26    <none>        9402/TCP   31m
service/cert-manager-webhook   ClusterIP   10.233.53.169   <none>        443/TCP    31m

NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cert-manager               1/1     1            1           31m
deployment.apps/cert-manager-cainjector    1/1     1            1           31m
deployment.apps/cert-manager-webhook       1/1     1            1           31m

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/cert-manager-7c869867bf               1         1         1       31m
replicaset.apps/cert-manager-cainjector-54c9d9b775    1         1         1       31m
replicaset.apps/cert-manager-webhook-7f7469bdb7       1         1         1       31m
```

Kserve components

```
root@est-selfservice-booking194-k8s-master-0:/home/infra/rappmanager/scripts/install# kubectl get all -n kserve
NAME                                         READY   STATUS    RESTARTS      AGE
pod/kserve-controller-manager-78c74d5b75-bqtls   2/2     Running   2 (95s ago)   28m

NAME                                             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/kserve-controller-manager-metrics-service   ClusterIP   10.233.56.191   <none>        8443/TCP   28m
service/kserve-controller-manager-service           ClusterIP   10.233.50.7     <none>        8443/TCP   28m
service/kserve-webhook-server-service               ClusterIP   10.233.22.190   <none>        443/TCP    28m

NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/kserve-controller-manager   1/1     1            1           28m

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/kserve-controller-manager-78c74d5b75   1         1         1       28m
```

ACM Components

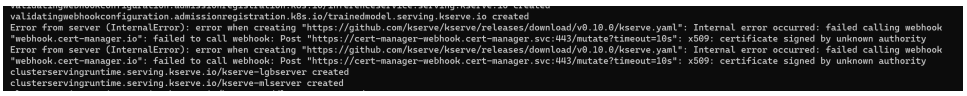
```
root@est-selfservice-booking194-k8s-master-0:/home/infra/rappmanager/scripts/install# kubectl get all -n default
NAME                                          READY   STATUS      RESTARTS   AGE
pod/mariadb-galera-0                          1/1     Running     0          35m
pod/policy-clamp-ac-a1pms-ppnt-5db94f4c89-4mdk4    1/1     Running     0          35m
pod/policy-clamp-ac-k8s-ppnt-587499d668-fsnnl      1/1     Running     0          35m
pod/policy-clamp-ac-kserve-ppnt-6dd9d49968-j7qjs   1/1     Running     0          35m
pod/policy-clamp-runtime-acm-84569dc946-67fn6      1/1     Running     0          35m
pod/policy-galera-config-vcvcw                0/1     Completed   0          35m
pod/policy-galera-init-hrxbf                  0/1     Completed   0          35m
pod/policy-models-simulator-788fbd465f-wr7wj       1/1     Running     0          35m

NAME                                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
service/aai-sim                       ClusterIP   10.233.16.82    <none>        6666/TCP                     35m
service/grpc-sim                      ClusterIP   10.233.60.235   <none>        6680/TCP                     35m
service/kubernetes                    ClusterIP   10.233.0.1      <none>        443/TCP                      78d
service/mariadb-galera                ClusterIP   10.233.18.204   <none>        3306/TCP                     35m
service/mariadb-galera-headless       ClusterIP   None            <none>        4567/TCP,4568/TCP,4444/TCP   35m
service/message-router                ClusterIP   10.233.47.23    <none>        3904/TCP                     35m
service/policy-clamp-ac-a1pms-ppnt    ClusterIP   10.233.38.44    <none>        8086/TCP                     35m
service/policy-clamp-ac-k8s-ppnt      NodePort    10.233.28.192   <none>        8083:30443/TCP               35m
service/policy-clamp-ac-kserve-ppnt   ClusterIP   10.233.33.99    <none>        8087/TCP                     35m
service/policy-clamp-runtime-acm      NodePort    10.233.46.213   <none>        6969:30442/TCP               35m
service/sdnc-sim                      ClusterIP   10.233.30.76    <none>        6668/TCP                     35m
service/so-sim                        ClusterIP   10.233.25.224   <none>        6669/TCP                     35m
service/vfc-sim                       ClusterIP   10.233.36.42    <none>        6670/TCP                     35m

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/policy-clamp-ac-a1pms-ppnt    1/1     1            1           35m
deployment.apps/policy-clamp-ac-k8s-ppnt      1/1     1            1           35m
deployment.apps/policy-clamp-ac-kserve-ppnt   1/1     1            1           35m
deployment.apps/policy-clamp-runtime-acm      1/1     1            1           35m
deployment.apps/policy-models-simulator       1/1     1            1           35m

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/policy-clamp-ac-a1pms-ppnt-5db94f4c89    1         1         1       35m
replicaset.apps/policy-clamp-ac-k8s-ppnt-587499d668      1         1         1       35m
replicaset.apps/policy-clamp-ac-kserve-ppnt-6dd9d49968   1         1         1       35m
replicaset.apps/policy-clamp-runtime-acm-84569dc946      1         1         1       35m
replicaset.apps/policy-models-simulator-788fbd465f       1         1         1       35m

NAME                               READY   AGE
statefulset.apps/mariadb-galera    1/1     35m

NAME                              COMPLETIONS   DURATION   AGE
job.batch/policy-galera-config    1/1           2m2s       35m
job.batch/policy-galera-init      1/1           66s        35m
```

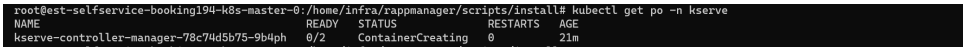
NONRTRIC Components

```
root@est-selfservice-booking194-k8s-master-0:/home/infra/rappmanager/scripts/install# kubectl get all -n nonrtric
NAME                                    READY   STATUS    RESTARTS   AGE
pod/a1-sim-osc-0                        1/1     Running   0          33m
pod/a1-sim-osc-1                        1/1     Running   0          32m
pod/a1-sim-std-0                        1/1     Running   0          33m
pod/a1-sim-std-1                        1/1     Running   0          32m
pod/a1-sim-std2-0                       1/1     Running   0          33m
pod/a1-sim-std2-1                       1/1     Running   0          32m
pod/a1controller-558776cc7b-7v44h       1/1     Running   0          33m
pod/capifcore-54c465899c-247md          1/1     Running   0          33m
pod/db-75c5789d97-7d2v4                 1/1     Running   0          33m
pod/dmeparticipant-6d4d5f9f98-xlgw4     1/1     Running   0          33m
pod/informationservice-0                1/1     Running   0          33m
pod/policymanagementservice-0           1/1     Running   0          33m
pod/rappmanager-0                       1/1     Running   0          33m

NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)             AGE
service/a1-sim                      ClusterIP   None            <none>        8085/TCP,8185/TCP   33m
service/a1controller                ClusterIP   10.233.35.135   <none>        8282/TCP,8383/TCP   33m
service/capifcore                   ClusterIP   10.233.46.221   <none>        8090/TCP            33m
service/dbhost                      ClusterIP   10.233.10.92    <none>        3306/TCP            33m
service/dmeparticipant              ClusterIP   10.233.48.158   <none>        8080/TCP            33m
service/informationservice          ClusterIP   10.233.43.62    <none>        9082/TCP,9083/TCP   33m
service/policymanagementservice     ClusterIP   10.233.62.226   <none>        9080/TCP,9081/TCP   33m
service/rappmanager                 ClusterIP   10.233.44.126   <none>        8080/TCP            33m
service/sdnctldb01                  ClusterIP   10.233.56.62    <none>        3306/TCP            33m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/a1controller        1/1     1            1           33m
deployment.apps/capifcore           1/1     1            1           33m
deployment.apps/db                  1/1     1            1           33m
deployment.apps/dmeparticipant      1/1     1            1           33m

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/a1controller-558776cc7b        1         1         1       33m
replicaset.apps/capifcore-54c465899c           1         1         1       33m
replicaset.apps/db-75c5789d97                  1         1         1       33m
replicaset.apps/dmeparticipant-6d4d5f9f98      1         1         1       33m

NAME                                      READY   AGE
statefulset.apps/a1-sim-osc               2/2     33m
statefulset.apps/a1-sim-std               2/2     33m
statefulset.apps/a1-sim-std2              2/2     33m
statefulset.apps/informationservice       1/1     33m
statefulset.apps/policymanagementservice  1/1     33m
statefulset.apps/rappmanager              1/1     33m
```

Troubleshooting

1. If Kserve installation failed or end up in the below state after "*install-all.sh*", Try run "*patch-kserve.sh*" script after "*install-all.sh*"





Uninstallation

To uninstall all the components

```
>./uninstall-all.sh
```